# *Supplementary Information*
# Predicting Solid State Material Platforms for Quantum Technologies

Oliver Lerstøl Hebnes
*Sopra Steria, Information Technology and Services, N-4020 Stavanger, Norway and*
*Department of Physics and Center for Computing in Science Education,*
*University of Oslo, N-0316 Oslo, Norway*

Marianne Etzelmüller Bathen*
*Advanced Power Semiconductor Laboratory, ETH Zürich, 8092 Zürich, Switzerland*

Øyvind Sigmundson Schøyen
*Department of Physics and Center for Computing in Science Education,*
*University of Oslo, N-0316 Oslo, Norway*

Sebastian G. Winther-Larsen
*Menon Economics, N-0369 Oslo, Norway and*
*Department of Physics and Center for Computing in Science Education,*
*University of Oslo, N-0316 Oslo, Norway*

Lasse Vines
*Department of Physics and Center for Materials Science and Nanotechnology,*
*University of Oslo, N-0316 Oslo, Norway*

Morten Hjorth-Jensen
*Department of Physics and Astronomy and Facility for Rare Ion Beams,*
*Michigan State University, East Lansing, MI 48824, USA and*
*Department of Physics and Center for Computing in Science Education,*
*University of Oslo, N-0316 Oslo, Norway*

**Contents:**



* bathen@aps.ee.ethz.ch

## SUPPLEMENTARY METHODS

### Featurization

To apply Matminer's featurization tools, we extend an existing implementation by Breuck *et al.* [1]. Table I contains an overview of the 39 chosen featurizers from Matminer. The featurization process results in 4876 physics informed descriptors.

The motivation behind the choice of featurizers is that we do not precisely know which features describe a suitable quantum host material. Therefore, we have collected a large quantity of descriptors in order to reliably predict potential materials for quantum technologies.

TABLE I: Descriptions of the 39 featurizers from Matminer that have been employed in this work. Descriptions are either found from Ref. [2] or from the project's Github page. For entries lacking references, we refer to Ref. [2].

| Features | Description | Reference |
|---|---|---|
| **Composition features** | | |
| AtomicOrbitals | Highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO) | [3] |
| AtomicPacking-Efficiency | Packing efficiency | [4] |
| BandCenter | Estimate absolute position of band center using geometric mean of electronegativity | [5] |
| ElementFraction | Fraction of each element in a composition | |
| ElementProperty | Statistics of various element properties | [6–8] |
| IonProperty | Maximum and average ionic character | [7] |
| Miedema | Formation enthalpies of intermetallic compounds, solid solutions, and amorphous phases using semi-empirical Miedema model | [9] |
| Stoichiometry | $L^p$ norm-based stoichiometric attributes | [7] |
| TMetalFraction | Fraction of magnetic transition metals | [8] |
| ValenceOrbital | Valence orbital attributes such as the mean number of electrons in each shell | [7] |
| YangSolid-Solution | Mixing thermochemistry and size mismatch terms | [10] |
| **Oxide composition features** | | |
| Electronegativity-Diff | Statistics on electronegativity difference between anions and cations | [8] |
| OxidationStates | Statistics of oxidation states | [8] |
| **Structure features** | | |
| DensityFeatures | Calculate density, volume per atom and packing fraction | - |
| GlobalSymmetry-Features | Determines spacegroup number, crystal system (1-7) and inversion symmetry | - |
| RadialDistribution-Function | Calculates the radial distribution function of a crystal system | - |
| CoulombMatrix | Generate the Coulomb matrix for nuclear interactions | [11] |
| PartialRadial-Distribution-Function | Compute the partial radial distribution | [12] |

| | | |
|---|---|---|
| | function of a crystal structure | |
| SineCoulomb-Matrix | Computes a variant of the coulomb matrix developed for periodic crystals | [13] |
| EwaldEnergy | Computes the energy from Coulombic interactions based on charge states of each site | [14] |
| BondFractions | Compute the fraction of each bond in a structure, based on nearest neighbors | [15] |
| Structural Heterogeneity | Calculates the variance in bond lengths and atomic volumes in a structure | [16] |
| MaximumPacking-Efficiency | Calculates the maximum packing efficiency of a structure | [16] |
| Chemical-Ordering | Computes how much the ordering of species differs from random in a structure | [16] |
| XRDPowder-Pattern | 1D array representing normalized powder diffraction of a structure as calculated by pymatgen | [6] |
| **Site features** | | |
| AGNI-Fingerprints | Calculates the product integral of RDF and Gaussian window function | [17] |
| AverageBond- Angle | Determines the average bond angle of a specific site with its nearest neighbors | [18] |
| AverageBond-Length | Determines the average bond length between one specific site and all its nearest neighbors | [18] |
| BondOrientational-Parameter | Calculates the averages of spherical harmonics of local neighbors | [19, 20] |
| ChemEnvSite-Fingerprint | Calculates the resemblance of given sites to ideal environment using pymatgens ChemEnv package | [21, 22] |
| Coordination-Number | The number of first nearest neighbors of a site | [22] |
| CrystalNN-Fingerprint | A local order parameter fingerprint for periodic crystals | - |
| GaussianSymm-Func | Calculates the gaussian radial and angular symmetry functions originally suggested for fitting machine learning potentials | [23, 24] |
| GeneralizedRadial-Distribution-Function | Computes the general radial distribution function for a site | [19] |
| LocalProperty-Difference | Computes the difference in elemental properties between a site and its neighboring sites | [16, 18] |
| OPSite-Fingerprint | Computes the local structure order parameters from a site's neighbor environment | [22] |
| Voronoi-Fingerprint | Calculates the Voronoi tessellation-based features around a target site | [25, 26] |
| **Density of state features** | | |
| DOSFeaturizer | Computes top contributors to the density of states at the VBM and CBM | [27] |
| **Band structure features** | | |
| BandFeaturizer | Converts a complex electronic band structure into discrete quantities | - |

### Optimization of machine learning methods

In the evaluation of the machine learning methods for the three different approaches, we apply a $5 \times 5$ stratified cross-validation when searching for the optimal hyperparameter combinations, see for example Ref. [28] for details on the cross-validation procedure. We apply four different evaluation metrics to each of the four machine learning algorithms discussed in this work.

The implementation of the machine learning methods was achieved through the machine learning library Scikit-Learn [29]. Adjusting many of the hyperparameters in Scikit-Learn resulted in severe overfitting for the machine learning methods random forests, gradient boosting, and decision trees. Therefore, most parameters are the default values defined by Scikit-Learn. The only parameter that was found to potentially improve the evaluation metric F1 (see Refs. [30, 31] for a definition of various metrics) was the maximum depth for the decision trees (these are also used as so-called weak learners in ensemble methods like gradient boosting and random forests [31]). In our work we adjusted the depth of the trees to vary from 1 to 8. For logistic regression, we chose to adjust the regularization strength with seven logarithmic values ranging from $10^{-3}$ to $10^{5}$, and used either 200 or 400 iterations to reach convergence.

When searching for the optimal number of principal components, we iterated over every odd number of principal components from 1 to the upper restricted number which defines an accumulated variance of 95 % from the principal component analysis. Due to a large number of principal components, we performed a fit of 25 folds for each of the 1232 parameter combinations, totaling up to 30.800 individual models.

#### *Ferrenti approach*

The grid search for the optimal number of principal components for the Ferrenti approach is visualized in Fig. 1, where we present the mean accuracy of the four ML models applied to the training set, and the balanced accuracy, precision, recall, and F1-score for the test set as a function of the number of principal components. For each principal component, we display the optimal combination of hyperparameters based on the F1-score.

In Table II, we include the precise measurements for each of the evaluation metrics for the optimal number of principal components. The optimal number is visualized as dotted lines in Fig. 1. The relevant hyperparameters for logistic regression were the maximum iterations. The number of iterations was set to 400, and the optimal regularization term was 0.46. For random forests and decision trees, we found the maximum depth to be 7. Gradient boosting, which normally uses a weak learner, had a decision tree depth of 4. We found the best performing method to be logistic

TABLE II: Optimal number of principal components and the respective scores (standard deviation) for each of the four ML methods logistic regression (LOG), decision trees (DT), random forests (RF) and gradient boosting (GB) in the Ferrenti approach, as visualized by the dash-dotted line in Fig. 1.

| Method | PC | Mean test | Mean precision | Mean recall | mean F1 |
|---|---|---|---|---|---|
| LOG | 171 | 0.98(0.012) | 0.98(0.011) | 0.99(0.007) | 0.99(0.007) |
| DT | 37 | 0.77(0.034) | 0.84(0.034) | 0.85(0.044) | 0.84(0.022) |
| RF | 53 | 0.87(0.027) | 0.88(0.022) | 0.98(0.010) | 0.93(0.014) |
| GB | 107 | 0.92(0.016) | 0.92(0.015) | 0.98(0.010) | 0.95(0.009) |

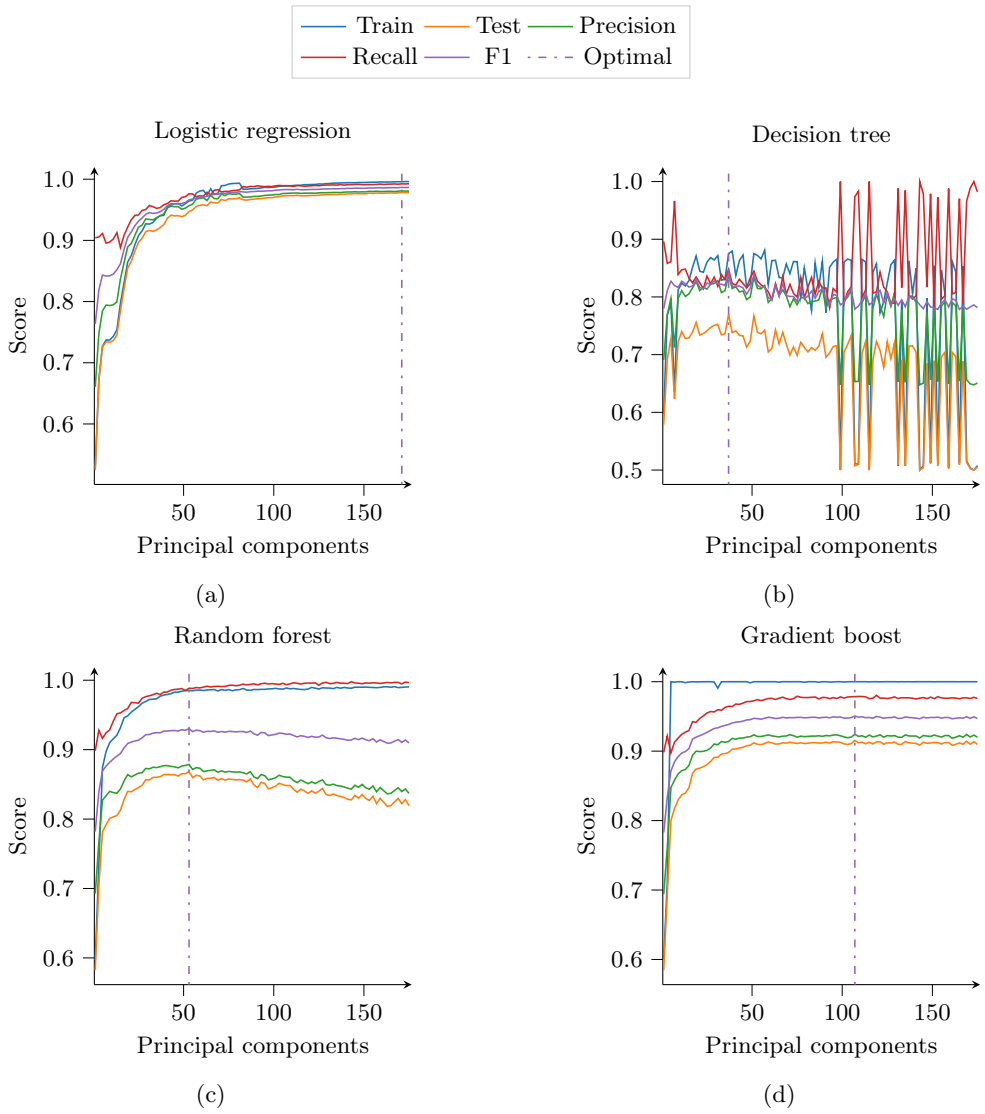


FIG. 1: Four figures displaying hyperparameter search for the Ferrenti approach. The best estimator is visualized for all hyperparameters as a function of principal components during a grid search with a $5 \times 5$ stratified cross-validation. The dotted lines mark the optimal hyperparameter-combination. Train stands for normal training accuracy, while test is the balanced accuracy on the test set. Precision, recall, and F1 scores are based on the test set. The number of principal components that explain the 95 % accumulated variance is 144, while the optimal model is found using the F1-score.

regression, but this method depends on a large number of principal components.

In Fig. 2, we visualize how the various machine learning methods interpret the principal components that are sorted in descending order by the explained variance, found through a $5 \times 5$

stratified cross-validation approach. To reach the 95 % accumulated explained variance, a total of 144 principal components were included. We have visualized the first 25 components since these capture the most important information. We note that most of the important features are found within the first five principal components.

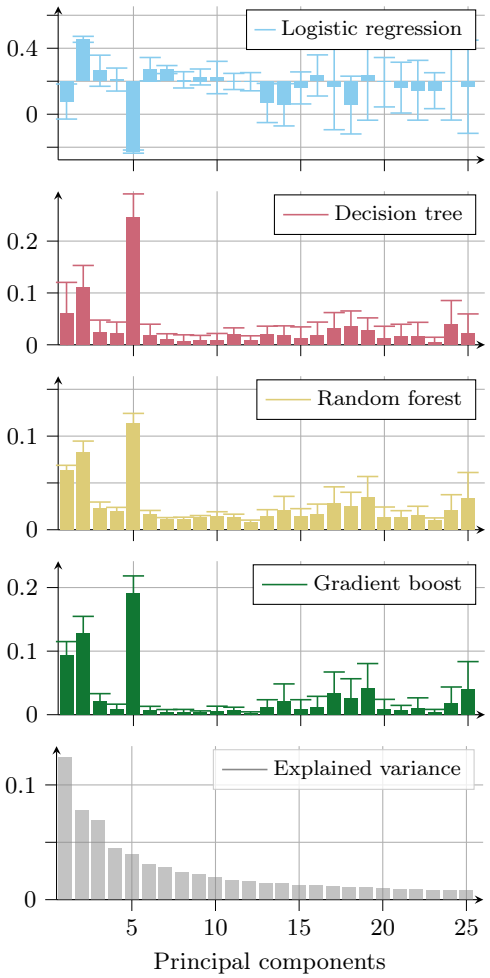


FIG. 2: Visualization of different parameters for the 25 most principal components ranked in descending order by the explained variance for the Ferrenti approach. The panels show the logistic regression coefficients, decision trees feature importance, random forests feature importance, gradient boosting feature importance, and explained variance that is retained by choosing each of the eigenvectors.

For logistic regression, we have visualized the mean fitted coefficients and the standard deviation in the top panel of Fig. 2. Large positive or negative coefficients can be considered increasingly important, where positive (negative) coefficients will contribute to making positive (negative) predictions. In the next three panels we visualize the mean impurity-based feature importance and the standard deviation for the ML methods decision trees, random forests, and gradient boosting,

TABLE III: Optimal number of principal components and the respective scores (standard deviation) for each of the four ML methods logistic regression (LOG), decision trees (DT), random forests (RF) and gradient boosting (GB) in the extended Ferrenti approach, as visualized by the dash-dotted line in Fig. 3.

| Method | PC | Mean test | Mean precision | Mean recall | mean F1 |
|---|---|---|---|---|---|
| LOG | 175 | 0.98(0.008) | 0.99(0.004) | 0.99(0.004) | 0.99(0.003) |
| DT | 25 | 0.69(0.034) | 0.86(0.015) | 0.93(0.021) | 0.90(0.008) |
| RF | 25 | 0.70(0.028) | 0.86(0.011) | 1.00(0.003) | 0.93(0.006) |
| GB | 93 | 0.85(0.025) | 0.93(0.011) | 0.99(0.004) | 0.96(0.007) |

respectively. We observe that the single most important feature for all ML methods in the Ferrenti approach is the fifth principal component. Selecting the highest values of this eigenvector, we find that the corresponding features originate from the DFT band gap of the elemental solids among the elements in the compound as calculated by Materials Agnostic Platform for Informatics and Exploration (MagPie).

After the first ten principal components, we observe that the methods adapt the other principal components with varying degrees. The coefficients for the case of logistic regression experience large fluctuations, but the three remaining models find that the first and second principal components are important in addition to the fifth. In order of importance, we observe that the second component's largest values correspond to the electronegativity, maximum ionic character, and covalent radius among the elements in the composition. The data originates from elemental calculations from MagPie and are aggregated as either minimum, mean, standard deviation, or maximum. While the first principal component encompasses the largest explained variance, it does not provide any specific information on which features it represents.

*Extended Ferrenti approach*

For the extended Ferrenti approach, the parameter grid search for principal components is visualized in Figure 3. All methods experience an almost perfect recall score for the first principal component due to the largely imbalanced dataset with 2141 suitable and 684 unsuitable candidates, which results in a ratio of 75 % : 25 % (suitable:unsuitable). This ratio comes as a consequence of the ML methods being able to correctly label many suitable candidates compared to the number of unsuitable candidates. On the other hand, we find a low precision score for the first component since the ML methods predict many materials as suitable regardless of whether they were labeled as suitable or unsuitable in the initial data labeling process. This trend is revealed when looking at the balanced accuracy score. For all figures, it remains the lowest score of the evaluation metrics largely due to the inaccuracy of true negatives for the cross-validations.

Overall the search for optimal hyperparameters in Fig. 3 for the extended Ferrenti approach bears resemblance to Fig. 1 for the Ferrenti approach. Logistic regression performs optimally for many principal components, and is the only method that continues to improve with an increasing number of components. The decision trees method exhibits a large fluctuation of scores, where the number of false positives is dominating the balanced accuracy score. The random forests method exhibits fewer fluctuations compared to decision trees as a consequence of the ensemble decision trees, while gradient boosting does not improve beyond 100 principal components.

The optimal hyperparameters for the extended Ferrenti approach are summarized in Table III.

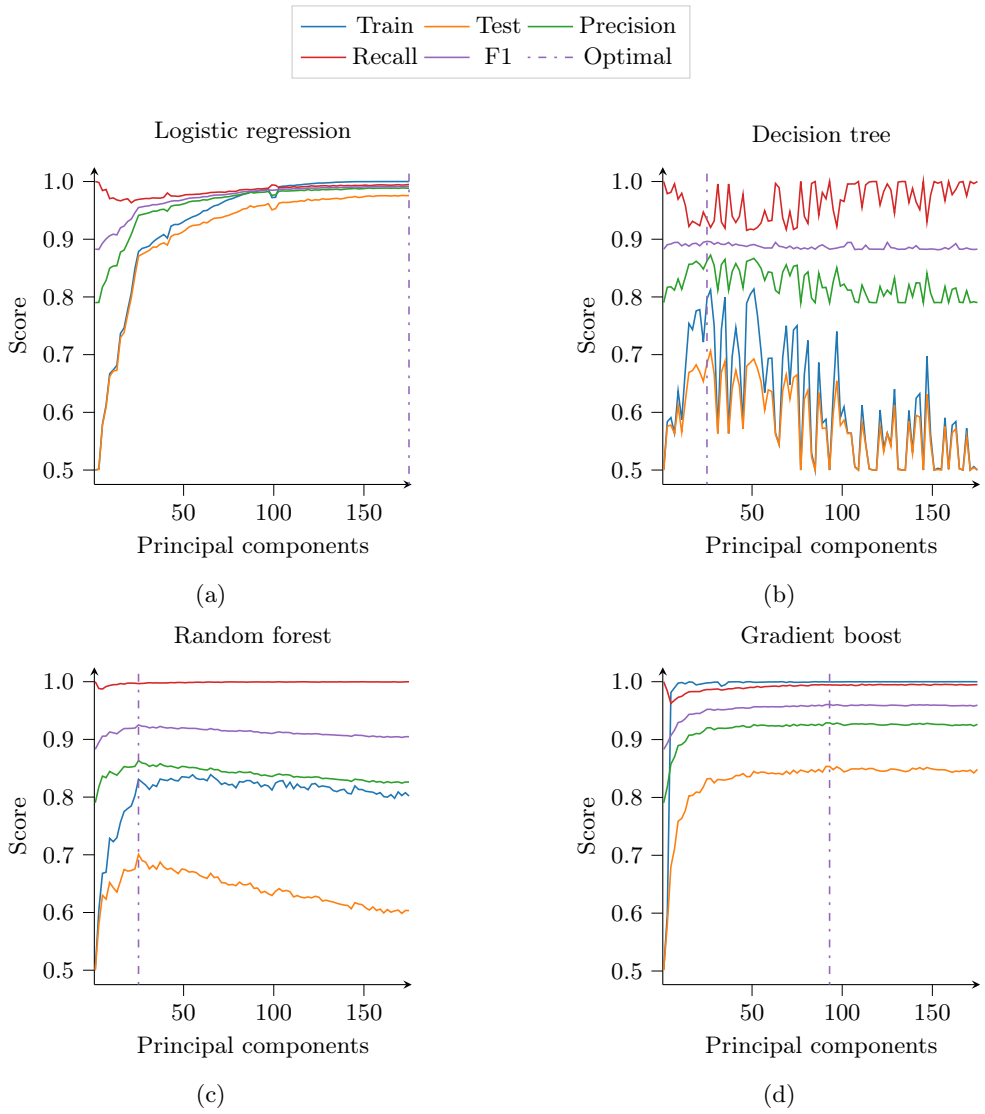


FIG. 3: Hyperparameter search for the extended Ferrenti approach. The best estimator is visualized for all hyperparameters as a function of principal components during a grid search with a $5 \times 5$ stratified cross-validation, and the dotted lines mark the optimal hyperparameter combination. Train stands for normal training accuracy, while test is the balanced accuracy on the test set. Precision, recall and F1 scores are based on the test set. The number of principal components that explain the 95 % accumulated variance is 159, while the optimal model is found using the F1-score.

We find that logistic regression with 175 principal components performs more or less like a perfect classifier with overall high scores. The decision trees and random forests methods have similar balanced accuracy scores with 0.69 and 0.70, respectively, due to challenges associated with predicting

true negative labels for 25 principal components. Lastly, we find that gradient boosting performs optimally at 93 principal components with a balanced accuracy score of 0.85.

The relevant hyperparameters for logistic regression are the regularization strength and the maximum iterations, which were set to 0.46 and 400, respectively. Smaller regularization values resulted in worse scores, while increasing values did not noticeably affect the results. The decision trees and random forests methods both found an optimal maximum depth of seven, where smaller values resulted in low precision but high recall. Therefore, the choice was made to facilitate a compromise between precision and recall. For gradient boosting, we find the optimal maximum depth to be four due to a decline in overall metrics for increasing depth. That is, except for the training accuracy, demonstrating a case of potential overfitting.

The interpretation of feature importance for the extended Ferrenti approach is substantially more difficult than in the Ferrenti approach. We find for logistic regression and decision trees that no feature is different than any other in the cross-validation due to a large variety of accuracy. However, we find that random forests and gradient boosting experience the fifth principal component as important. Similar to the Ferrenti approach, the extended Ferrenti approach finds the DFT band gap of the elemental solids among the elements in the composition to be important. The feature regarding the band gap originates from the highest value from the first principal component. This means that, if we consider, e.g., the compound SiC, the band gaps of both SiC, Si and C would be considered important by the ML methods.

### *Empirical approach*

Lastly, we turn to the data labeling using the empirical approach, which resulted in 404 unsuitable and 202 suitable candidates through the initial selection process. However, in contrast to the two labeled data sets discussed above, the majority of the entries was in this case labeled as unsuitable candidates.

The grid search for the optimal number of principal components is visualized in Fig. 4 for the empirical approach. Interestingly, we find that all ML methods experience high scores for just a few principal components, where the first principal component earns at least 0.93 scores for all evaluation metrics.

Logistic regression experiences improvement of all scores for an increasing number of principal components, yet only up 5 % in scores compared to the one-dimensional representation of one principal component. Thus, one can argue whether the increase in performance is worthwhile, considering a one-dimensional representation with just a few percentage losses of performance. However, with multiple principal components, we find the largest increase in precision, which is a sign that the one-dimensional representation tends to wrongly predict candidates as suitable when they are in fact unsuitable. Decision trees and random forests exhibit the best performance for just a few principal components, and experience a considerable degree of overfitting for larger values. Gradient boosting, in contrast to the case for the two Ferrenti approaches, also experiences the best performance for a few principal components.

The optimal hyperparameters are summarized in Table IV, where all ML methods exhibit high evaluation metrics. The difference in the number of optimal principal components is prominent. Logistic regression finds an optimum at 61 with an F1-score of 0.98. Decision trees uses only 9 principal components to achieve an F1 score of 0.95, while random forests needs 27 principal components to gain an F1 score of 0.97. Gradient boosting performs optimally at 13 principal components with a mean F1-score of 0.96. The relevant hyperparameters were the regularization term for logistic regression, which was set to 0.021, and the maximum number of iterations at 400. The decision trees uses a maximum depth of 6, where larger values only increased the training

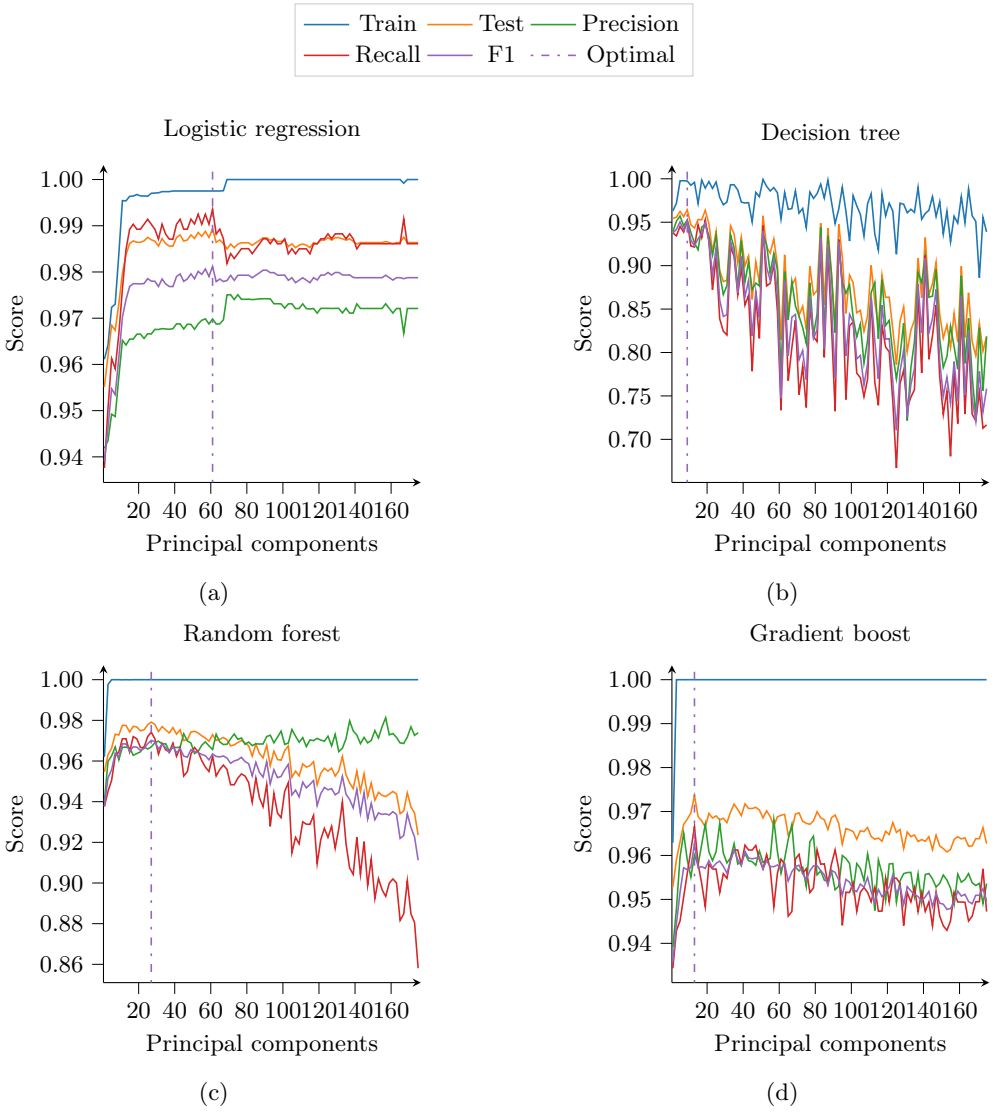


FIG. 4: Hyperparameter search for the empirical approach. The best estimator is visualized for all hyperparameters as a function of principal components during a grid search with a $5 \times 5$ stratified cross-validation, and the dotted lines mark the optimal hyperparameter-combination. Train stands for normal training accuracy, while test is the balanced accuracy on the test set. Precision, recall, and F1-scores are based on the test set. The number of principal components that explain the 95 % accumulated variance is 103, while the optimal model is found using the F1-score.

accuracy. The maximum depth for random forests was set to 6. For gradient boosting, which uses a weak learner, the depth of the trees was set to 4.

In Fig. 5, we visualize how the four ML methods in the empirical approach interpret the principal components that are sorted in descending order by the explained variance, found through a $5 \times 5$

TABLE IV: Optimal number of principal components and the respective scores (standard deviation) for each of the four ML methods logistic regression (LOG), decision trees (DT), random forests (RF) and gradient boosting (GB) in the empirical approach, as visualized by the dash-dotted line in Fig. 4.

| Method | PC | Mean test | Mean precision | Mean recall | mean F1 |
|---|---|---|---|---|---|
| LOG | 61 | 0.99(0.011) | 0.97(0.032) | 0.99(0.016) | 0.98(0.018) |
| DT | 9 | 0.96(0.019) | 0.95(0.040) | 0.95(0.033) | 0.95(0.026) |
| RF | 27 | 0.98(0.020) | 0.97(0.033) | 0.97(0.031) | 0.97(0.026) |
| GB | 13 | 0.97(0.016) | 0.96(0.036) | 0.97(0.029) | 0.96(0.022) |

stratified cross-validation approach. Here, the most important information is captured by the first 15 components. The most important features seem to be found within the first few principal components. Logistic regression is represented in the top panel of Fig. 5 where the mean fitted coefficients and the standard deviation are visualized. In the three next panels, namely for the ML methods decision trees, random forests, and gradient boosting, we visualize the mean impurity-based feature importance, along with the standard deviation. We observe that the single most important feature for all ML methods is the first principal component in the case of the empirical approach. Selecting the highest values of this eigenvector, we find that the corresponding features is a complex combination of several material properties, including bond orientational parameters, coordination numbers, and the radial distribution function of a compound's crystal system.

## SUPPLEMENTARY RESULTS

### Prediction statistics

Figure 6 provides a visualization of the number of predicted material candidates from the (a) Ferrenti, (b) extended Ferrenti and (c) empirical approaches. Suitable and unsuitable candidates are marked in green and red, respectively. We observe that the number of predicted suitable candidates is similar for the Ferrenti and extended Ferrenti approaches, albeit higher in the case of the latter, for all four ML methods, while the empirical approach results in a substantially narrower selection of contender materials. Note that the confidence threshold was set to 0.5 in this case for all the methods and approaches.

For the Ferrenti approach, logistic regression finds a total of 12.380 suitable candidates, while decision trees is the most conservative with 11.315. Random forests has the most optimistic estimate with 14.278, while gradient boosting finds 11.835 suitable candidates. The four ML methods agree on 6804 suitable candidates, however, many of the materials are predicted with a confidence similar to that of a coin-flip. If we were to raise the minimum bar of a prediction to 0.75, the four methods would only agree on 1784 suitable candidates.

The perhaps more liberal extended Ferrenti approach yields the largest number of predicted candidates with 14.993, 14.407, 15.351 and 13.788 for logistic regression, decision trees, random forests, and gradient boosting, respectively. Due to the less stringent restrictions compared to the Ferrenti approach, we find a large number (9227) of entries that the four ML methods agree on.

The four ML methods predict radically fewer suitable candidates for the empirical approach as compared to the two former approaches, where only 842, 1197, 543, and 596 materials are predicted as suitable by logistic regression, decision trees, random forests, and gradient boosting,

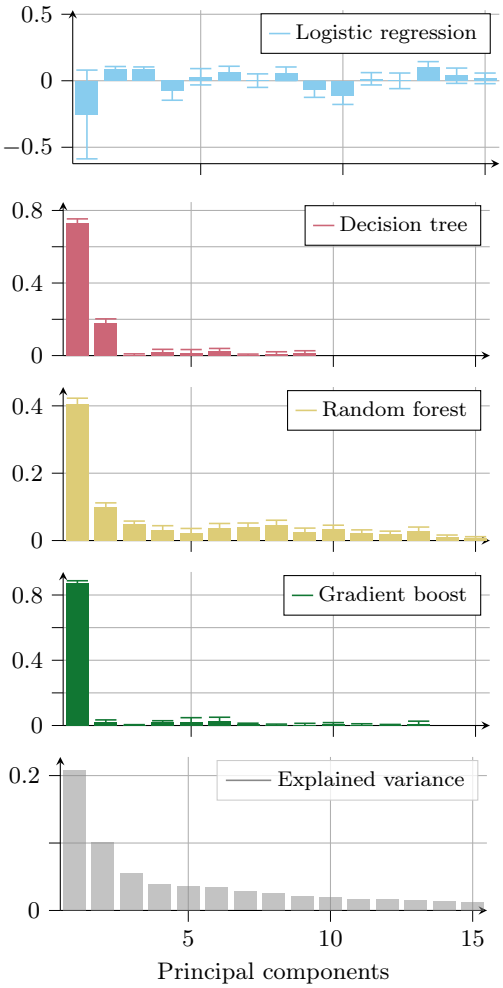


FIG. 5: Visualization of different parameters for the 15 most principal components ranked in descending order by the explained variance for the empirical approach. The panels show the logistic regression coefficients, decision trees feature importance, random forests feature importance, gradient boosting feature importance, and explained variance that is retained by including each of the eigenvectors.

respectively. The large majority of the unsuitable candidates are predicted with high probability except for random forests due to the ensemble of trees. All methods, however, agree on 214 suitable candidates above a 0.5 threshold limit.

### Predicted materials

Table V displays the 66 predicted candidate materials that all four machine learning methods, using the training and test sets derived in the empirical approach, agreed on with a confidence

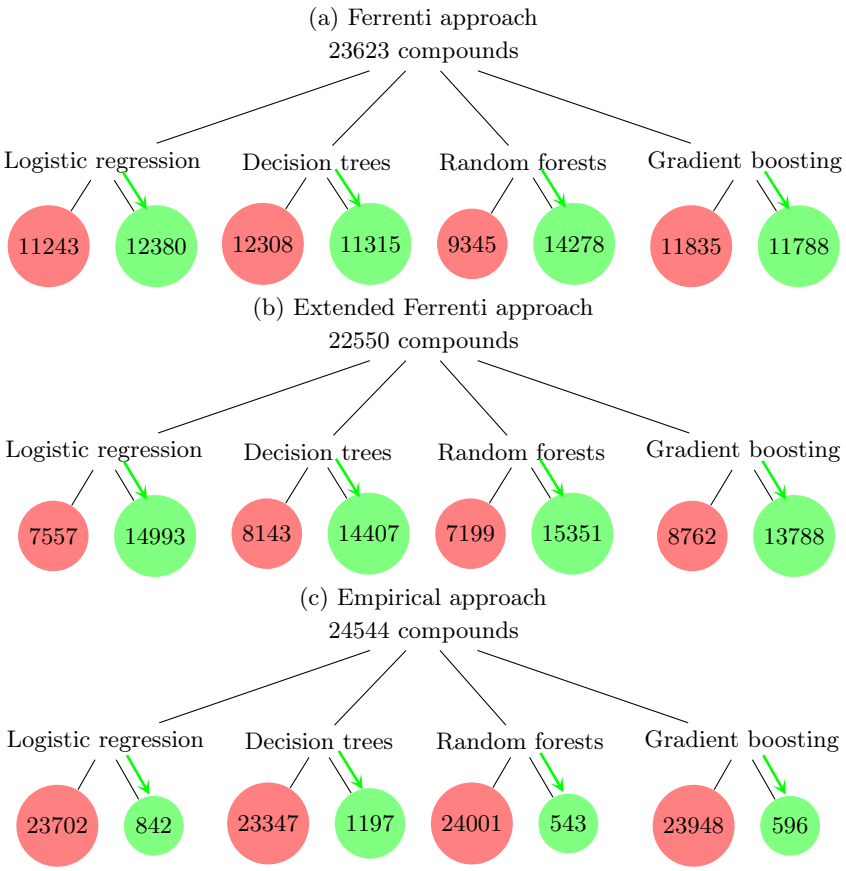


FIG. 6: Visualization of the number of predicted material candidates from the (a) Ferrenti, (b) extended Ferrenti and (c) empirical approaches. Green and red nodes represent suitable and unsuitable candidates, respectively.

cut-off set to 0.75. All band gaps are taken from the Materials Project database as calculated using DFT and the PBE functional. Note that materials can appear several times in the list due to different structures of the same composition. The list contains 5 elementary (unary), 46 binary and 15 ternary compounds.

TABLE V: The 66 predicted candidates that all models in the empirical approach agreed on to a 0.75 thereshold. All band gaps were taken from the Materials Project (MP) database, and materials can appear several times in the list due to different structures. The list contains 5 elementary (unary), 46 binary and 15 ternary compounds.

| Compound formula | MP ID | Band gap from MP (eV) |
|---|---|---|
| Ge | mp-137 | 0.87 |
| CdTe | mp-406 | 1.22 |
| HgSe | mp-820 | 0.12 |
| GeTe | mp-938 | 0.82 |
| MgTe | mp-1039 | 2.36 |
| CdSe | mp-1070 | 0.55 |
| GaSb | mp-1156 | 0.36 |
| BP | mp-1479 | 1.46 |
| $MoSe_2$ | mp-1634 | 1.41 |
| BN | mp-1639 | 4.64 |
| YbTe | mp-1779 | 1.52 |
| SnS | mp-1876 | 0.95 |
| SnTe | mp-1883 | 0.66 |
| GeTe | mp-2612 | 0.61 |
| AlSb | mp-2624 | 1.26 |
| CdSe | mp-2691 | 0.50 |
| SnSe | mp-2693 | 0.82 |
| $CdSnAs_2$ | mp-3829 | 0.30 |
| $GaCuTe_2$ | mp-3839 | 0.55 |
| $ZnGeAs_2$ | mp-4008 | 0.56 |
| $ZnGeP_2$ | mp-4524 | 1.20 |
| $GaAgTe_2$ | mp-4899 | 0.19 |
| $CdSnP_2$ | mp-5213 | 0.67 |
| $GaCuS_2$ | mp-5238 | 0.70 |
| SnS | mp-10013 | 0.23 |
| BAs | mp-10044 | 1.25 |
| GeSe | mp-10759 | 0.44 |
| MgSe | mp-10760 | 1.97 |
| CdTe | mp-12779 | 0.61 |
| MgSe | mp-13031 | 2.54 |
| MgTe | mp-13033 | 2.31 |
| TePb | mp-19717 | 1.05 |
| InAs | mp-20305 | 0.30 |
| InP | mp-20351 | 0.46 |
| $InAgSe_2$ | mp-20554 | 0.36 |
| InN | mp-22205 | 0.47 |
| AgI | mp-22894 | 1.39 |
| CuI | mp-22895 | 1.17 |
| CuBr | mp-22913 | 0.48 |
| CuCl | mp-22914 | 0.80 |
| AgI | mp-22919 | 1.00 |
| AgI | mp-22925 | 1.72 |
| Br | mp-23154 | 1.32 |
| TlI | mp-23197 | 2.25 |
| AgBr | mp-23231 | 0.79 |
| $BC_2N$ | mp-30148 | 2.10 |

| | | |
|---|---|---|
| CuI | mp-569346 | 1.21 |
| Hg | mp-569360 | 0.22 |
| $Ga_2Os$ | mp-570875 | 0.66 |
| $BC_2N$ | mp-629458 | 1.84 |
| InP | mp-966800 | 0.51 |
| GeC | mp-1002164 | 1.84 |
| TlP | mp-1007776 | 0.12 |
| $BC_2N$ | mp-1008523 | 1.64 |
| BP | mp-1008559 | 1.07 |
| OsC | mp-1009540 | 0.17 |
| SiSn | mp-1009813 | 0.41 |
| $ZnCdSe_2$ | mp-1017534 | 1.85 |
| MgSe | mp-1018040 | 2.57 |
| AlSb | mp-1018100 | 0.91 |
| AlBi | mp-1018132 | 0.30 |
| Ge | mp-1067619 | 0.79 |
| $Ga_2Ru$ | mp-1072429 | 0.12 |
| $ZnCd_3Se_4$ | mp-1078597 | 1.72 |
| $BC_2N$ | mp-1079201 | 1.17 |
| Ge | mp-1198022 | 0.67 |

Table VI displays the 47 predicted candidates that all four machine learning methods and all three approaches (Ferrenti, extended Ferrenti and empirical) agreed on (to a 0.5 threshold level). The list contains 8 elemental, 29 binary, and 10 tertiary compounds.

TABLE VI: The 47 predicted candidates that all four ML models and all three approaches agree on as suitable. All band gaps were taken from the Materials Project (MP) database, and materials can appear several times in the list due to different structures. The list contains 8 elemental, 29 binary, and 10 tertiary compounds.

| Compound formula | MP ID | Band gap from MP (eV) |
|---|---|---|
| P | mp-157 | 7.47 |
| SiRu | mp-189 | 0.63 |
| BN | mp-344 | 0.41 |
| HgSe | mp-820 | 3.96 |
| FeSi | mp-871 | 2.06 |
| MgTe | mp-1039 | 6.62 |
| CdSe | mp-1070 | 3.91 |
| BP | mp-1479 | 2.89 |
| CdSe | mp-2691 | 2.40 |
| $ZnSiAs_2$ | mp-3595 | 1.57 |
| $ZnGeAs_2$ | mp-4008 | 5.93 |
| $CdSnP_2$ | mp-5213 | 5.15 |
| $Si_2Mo$ | mp-8938 | 0.84 |
| BAs | mp-10044 | 0.24 |
| GeSe | mp-10759 | 2.07 |
| $N_2$ | mp-12103 | 0.51 |
| $BeSiN_2$ | mp-15704 | 3.85 |
| InP | mp-20351 | 0.24 |

| | | |
|---|---|---|
| InN | mp-22205 | 1.53 |
| AgCl | mp-22922 | 0.47 |
| I | mp-23153 | 4.61 |
| Br | mp-23154 | 4.59 |
| TlI | mp-23197 | 2.86 |
| AgBr | mp-23231 | 0.43 |
| $H_2$ | mp-23907 | 7.50 |
| $Ge_3As_4$ | mp-569600 | 2.43 |
| TlCl | mp-569639 | 2.09 |
| $Sn_3As_4$ | mp-570377 | 5.57 |
| $H_2$ | mp-634659 | 1.08 |
| $N_2$ | mp-672234 | 0.42 |
| $TiFe_2Ge$ | mp-866375 | 2.64 |
| InP | mp-966800 | 5.76 |
| GeC | mp-1002164 | 1.42 |
| $B_2AsP$ | mp-1008528 | 1.39 |
| BP | mp-1008559 | 1.53 |
| $BeSiAs_2$ | mp-1009087 | 0.13 |
| OsC | mp-1009540 | 2.46 |
| ScP | mp-1009746 | 3.48 |
| SiSn | mp-1009813 | 0.84 |
| SnC | mp-1009820 | 2.28 |
| AlBi | mp-1018132 | 2.78 |
| $Al_3BN_4$ | mp-1019380 | 1.10 |
| GeRu | mp-1025397 | 4.54 |
| PbS | mp-1057015 | 3.46 |
| Ge | mp-1067619 | 0.19 |
| $ZnCd_3S_4$ | mp-1078780 | 2.38 |
| $Ga_4BiAs_3$ | mp-1079228 | 3.37 |

**Feature analysis**

Figure 7 displays the number of predicted suitable materials for all approaches (all ML methods in an approach agree to a 0.75 confidence level) as a function of the (a) covalent radius of the elements in the composition, (b) maximum packing efficiency, and standard deviation of the radial distribution function (RDF) with center at (c) 2.0 and (d) 3.0. The corresponding figure in the main text shows the standard deviation of the RDF with Gaussian center at 1.0.

We observe that the covalent radii of the materials (see Fig. 7a) distribute with two peaks in the data in all approaches. The maxima are slightly shifted towards the right for the empirical approach compared to the two others but otherwise the data distributions are similar. The trend of two data peaks is repeated for the maximum packing efficiency (see Fig. 7b) but is much more prominent for the empirical approach. This indicates that the material density, or in other words the bond length, is an important parameter for QT suitability.

Figure 8 shows the number of predicted suitable materials for all approaches (all ML methods in an approach agree to a 0.75 confidence level) as a function of bond orientational parameters and band gap. Panels (a) chemical environment site fingerprint, (b) crystal fingerprint and (c) OP site fingerprint are related to bond orientational parameters, while (d) shows the material band gaps referenced to the normalized mean of the entire data set.

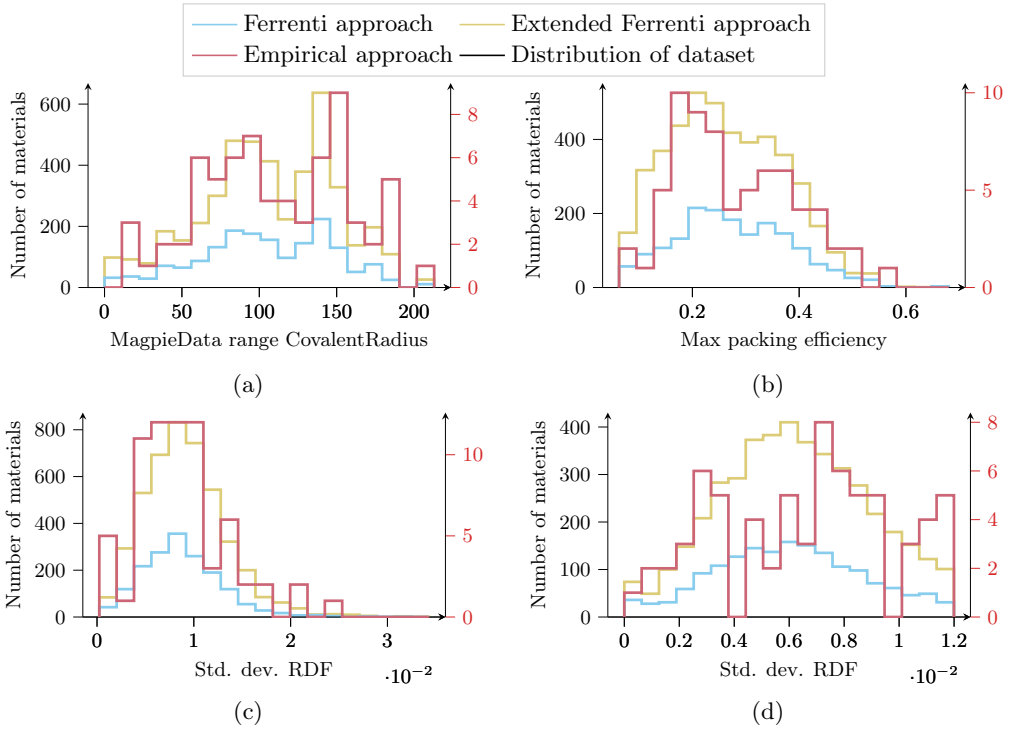


FIG. 7: Number of predicted suitable materials as a function of the (a) covalent radius, (b) maximum packing efficiency, and standard deviation of the radial distribution function with center at (c) 2.0 and (d) 3.0. The Ferrenti and extended Ferrenti approaches refer to the left y-axis and the empirical approach to the right. All panels were taken for a 0.75 cut-off.


[1] P.-P. D. Breuck, M. L. Evans, and G.-M. Rignanese, Robust model benchmarking and bias-imbalance in data-driven materials science: a case study on MODNet, Journal of Physics: Condensed Matter **33**, 404002 (2021).

[2] L. Ward, A. Dunn, A. Faghaninia, N. E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K. A. Persson, G. J. Snyder, I. Foster, and A. Jain, Matminer: An open source toolkit for materials data mining, Computational Materials Science **152**, 60 (2018).

[3] S. Kotochigova, Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark, Local-density-functional calculations of the energy of atoms, Physical Review A **55**, 191 (1997).

[4] K. J. Laws, D. B. Miracle, and M. Ferry, A predictive structural model for bulk metallic glasses, Nature Communications **6** (2015).

[5] M. A. Butler and D. S. Ginley, Prediction of flatband potentials at semiconductor-electrolyte interfaces from atomic electronegativities, Journal of The Electrochemical Society **125**, 228 (1978).

[6] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science **68**, 314 (2013).

[7] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, A general-purpose machine learning framework

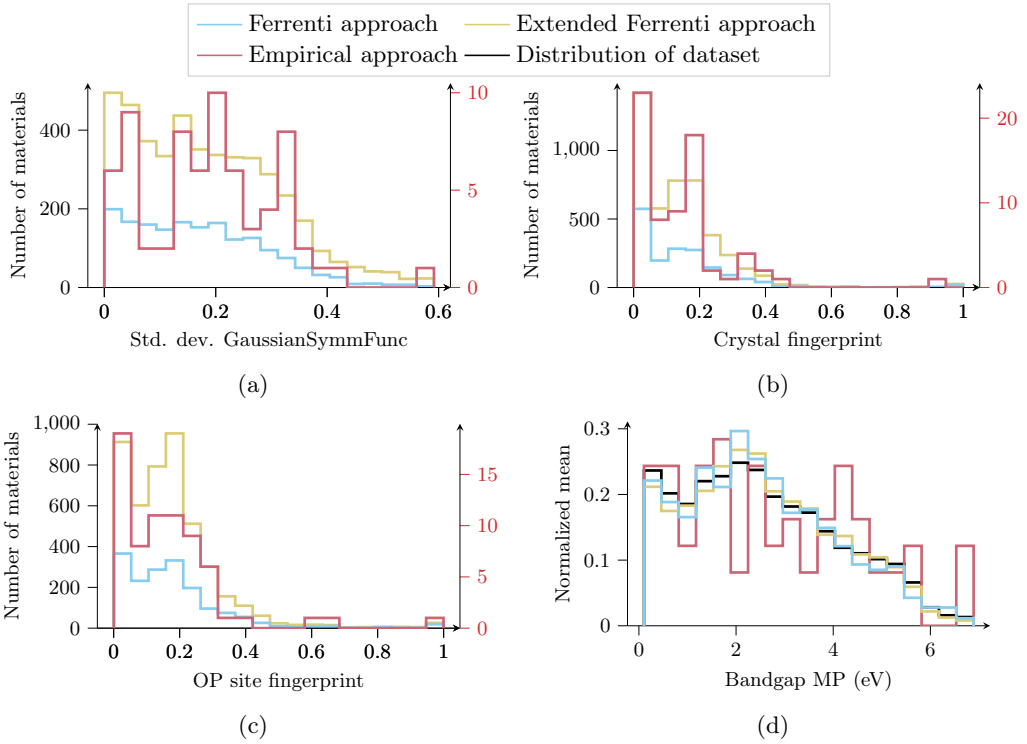


FIG. 8: Number of predicted suitable materials as a function of the (a) standard deviation of the chemical environment of the gaussian symmetric function. Panels (b) crystal fingerprint and (c) OP site fingerprint are related to bond orientational parameters, while (d) shows the band gap referenced to the normalized mean of the entire data set. The Ferrenti and extended Ferrenti approaches refer to the left y-axis and the empirical approach to the right. All panels were taken for a 0.75 cut-off.

for predicting properties of inorganic materials, npj Computational Materials **2** (2016).

[8] A. M. Deml, R. O'Hayre, C. Wolverton, and V. Stevanović, Predicting density functional theory total energies and enthalpies of formation of metal-nonmetal compounds by linear regression, Physical Review B **93**, 8 (2016).

[9] A. W. Weeber, Application of the miedema model to formation enthalpies and crystallisation temperatures of amorphous alloys, Journal of Physics F: Metal Physics **17**, 809 (1987).

[10] X. Yang and Y. Zhang, Prediction of high-entropy stabilized solid-solution in multi-component alloys, Materials Chemistry and Physics **132**, 233 (2012).

[11] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning, Physical Review Letters **108** (2012).

[12] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, How to represent crystal structures for machine learning: Towards fast prediction of electronic properties, Physical Review B **89**, 20 (2014).

[13] F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, Crystal structure representations for machine learning models of formation energies, International Journal of Quantum Chemistry **115**, 1094 (2015).

[14] P. P. Ewald, Die berechnung optischer und elektrostatischer gitterpotentiale, Annalen der Physik **369**,

253 (1921).
[15] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space, The Journal of Physical Chemistry Letters **6**, 2326 (2015).
[16] L. Ward, R. Liu, A. Krishna, V. I. Hegde, A. Agrawal, A. Choudhary, and C. Wolverton, Including crystal structure attributes in machine learning models of formation energies via voronoi tessellations, Physical Review B **96**, 2 (2017).
[17] V. Botu and R. Ramprasad, Adaptive machine learning framework to accelerate ab initio molecular dynamics, International Journal of Quantum Chemistry **115**, 1074 (2014).
[18] M. de Jong, W. Chen, R. Notestine, K. Persson, G. Ceder, A. Jain, M. Asta, and A. Gamst, A statistical learning framework for materials science: Application to elastic moduli of k-nary inorganic polycrystalline compounds, Scientific Reports **6**, 1 (2016).
[19] A. Seko, H. Hayashi, K. Nakayama, A. Takahashi, and I. Tanaka, Representation of compounds for machine-learning prediction of physical properties, Physical Review B **95** (2017).
[20] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, Bond-orientational order in liquids and glasses, Physical Review B **28**, 784 (1983).
[21] D. Waroquiers, X. Gonze, G.-M. Rignanese, C. Welker-Nieuwoudt, F. Rosowski, M. Göbel, S. Schenk, P. Degelmann, R. André, R. Glaum, and G. Hautier, Statistical analysis of coordination environments in oxides, Chemistry of Materials **29**, 8346 (2017).
[22] N. E. R. Zimmermann, M. K. Horton, A. Jain, and M. Haranczyk, Assessing local structure motifs using order parameters for motif recognition, interstitial identification, and diffusion path characterization, Frontiers in Materials **4** (2017).
[23] J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, The Journal of Chemical Physics **134**, 074106 (2011).
[24] A. Khorshidi and A. A. Peterson, Amp: A modular approach to machine learning in atomistic simulations, Computer Physics Communications **207**, 310 (2016).
[25] H. L. Peng, M. Z. Li, and W. H. Wang, Structural signature of plastic deformation in metallic glasses, Physical Review Letters **106** (2011).
[26] Q. Wang and A. Jain, A transferable machine-learning framework linking interstice distribution and plastic heterogeneity in metallic glasses, Nature Communications **10** (2019).
[27] M. T. Dylla, A. Dunn, S. Anand, A. Jain, and G. J. Snyder, Machine learning chemical guidelines for engineering electronic structures in half-heusler thermoelectric materials, Research **2020**, 1 (2020).
[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction* (Springer Verlag, Berlin, 2009).
[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, Scikit-learn: Machine learning in python, Journal of machine learning research (2011).
[30] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning* (Springer, Boston, MA, 2010).
[31] A. Geron, *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow, 3rd edition* (O'Reilly, New York, 2022).